

A decorative graphic on the left side of the page, consisting of overlapping blue and grey geometric shapes that form a large, stylized arrow pointing to the right.

XDataSet5 JAVA API Reference

VERSION 23.03.14.1

서울특별시 구로구 디지털로 272, 1110 (구로동, 한신 IT 타워)

Phone 02-2108-8030 • Fax 02-2108-8031

www.softbase.co.kr

Copyright © 2010 SOFTBase Inc. All rights reserved

목차

XDataSet Class 개요	5
Constructor	5
Methods	6
XDataSet API 설명	8
getColumnCount	8
getColumnNames.....	9
getColumnNamesArray.....	10
getData.....	11
getDataSetNames.....	13
getDataSetNamesArray	14
getDataSetType	15
getDoubleData	16
getFloatData	18
getFullVersion	20
getIntData	21
getLongData	23
getOutputColumnNames.....	25
getOutputColumnNamesArray.....	26
getOutputData.....	27
getOutputRecordCount	28
getRecordCount	29
getRecordKey	30
getRecordType.....	31
getReturnByteArrayData.....	34
getReturnStringData	35
getScreenNo	36
getTerminalIpAddress.....	37
getTransactionCode	38
getTransactionHeader	39
getTransactionMapId.....	40
getUserHeader.....	41
getXDataSetOutputDataString	42
getVersion.....	43
isDeleteRecord.....	44
isInsertRecord.....	45
isUpdateRecord	46

returnData.....	47
returnPartData	48
setCharacterSet.....	50
setColumnNames	52
setData	54
setDataSetName	55
setDataSetNames	56
setDoubleData	57
setErrorMessage.....	58
setFloatData	59
setIntData.....	60
setLongData.....	61
setMessage.....	62
setHttpResponse.....	63
setRecordKey.....	64
setUserHeader.....	65
replaceHTMLTagFilter.....	66

XDataSet Class 개요

Constructor

이름	설명
XDataSet5 (HttpRequest, HttpResponse)	<ul style="list-style-type: none"> 하나의 Page에서 Request/Response를 처리하는 경우 사용한다. HttpRequest 및 HttpResponse 오브젝트를 파라미터로 전달받아서 데이터 파싱 작업을 자동으로 수행한다. 출력 데이터를 설정하기 위해서는 데이터 셋 이름 및 데이터 셋의 칼럼 이름이 자동으로 관리되므로 별도로 설정을 할 필요가 없다.
XDataSet5 (String)	<ul style="list-style-type: none"> Request만을 처리하는 경우에 사용한다. 입력 데이터에 대한 데이터 파싱 작업을 파라미터로 넘겨오는 데이터를 이용해서 자동으로 수행한다. 파라미터는 "XDATASET5"키에 대한 값이다. 출력 데이터를 전송하기 위해서는 SetHttpResponse 함수를 사용해서 HttpResponse 오브젝트를 xDataSet 오브젝트에 설정해야 한다. 출력 데이터를 설정하기 위해서는 데이터 셋 이름 및 데이터 셋의 칼럼 이름이 자동으로 관리되므로 별도로 설정을 할 필요가 없다.
XDataSet5 (HttpResponse)	<ul style="list-style-type: none"> Response만을 처리하는 경우에 사용한다. 출력 데이터를 설정하기 위해서는 데이터 셋 이름 및 데이터 셋의 칼럼 이름을 설정해야 한다.
XDataSet5 ()	<ul style="list-style-type: none"> Response만을 처리하는 경우에 사용한다. 출력 데이터를 전송하기 위해서는 SetHttpResponse 함수를 사용해서 HttpResponse 오브젝트를 설정해야 한다. 출력 데이터를 설정하기 위해서는 데이터 셋 이름 및 데이터 셋의 칼럼 이름을 설정해야 한다.

Methods

이름	설명
getColumnCount	▪ Request에 대한 데이터 셋의 칼럼 수를 리턴한다.
getColumnNames	▪ Request에 대한 데이터 셋의 칼럼 이름들을 Vector로 리턴한다
getColumnNamesArray	▪ Request에 대한 데이터 셋의 칼럼 이름들을 문자열 Array로 리턴한다
getData	▪ Request에 대한 데이터셋의 데이터를 얻어온다.
getDataSetNames	▪ Request에 대한 데이터 셋들의 이름을 Vector로 리턴한다
getDataSetNamesArray	▪ Request에 대한 데이터 셋들의 이름을 문자열 Array로 리턴한다
getDataSetType	▪ 데이터셋의 입출력 구분 정보를 반환한다.
getDoubleData	▪ Request에 대한 데이터의 데이터를 double 형으로 얻어온다.
getFloatData	▪ Request에 대한 데이터의 데이터를 float 형으로 얻어온다.
getFullVersion	▪ XDataSet 라이브러리의 Full Version 정보를 얻어온다.
getIntData	▪ Request에 대한 데이터의 데이터를 int 형으로 얻어온다.
getLongData	▪ Request에 대한 데이터의 데이터를 long 형으로 얻어온다.
getOutputColumnNames	▪ 출력용 데이터 셋의 칼럼 이름들을 Vector로 리턴한다.
getOutputColumnNamesArray	▪ 출력용 데이터셋 이름 배열을 리턴한다.
getOutputColumnCount	▪ 출력용 데이터셋의 칼럼 수를 리턴한다.
getOutputData	▪ 출력용 데이터셋에 설정된 데이터를 리턴한다.
getOutputRecordCount	▪ 출력용 데이터셋의 레코드 수를 리턴한다.
getRecordCount	▪ Request에 대한 데이터셋의 레코드 수를 리턴한다.
getRecordKey	▪ Request에 대한 데이터 셋의 Record의 키를 리턴한다.
getRecordType	▪ Request에 대한 데이터 셋 레코드의 유형을 리턴한다.
getReturnByteArrayData	▪ 출력용 리턴 데이터를 바이트 배열 형식으로 리턴한다.
getReturnStringData	▪ 출력용 리턴 데이터를 문자열 형식으로 리턴한다.
getScreenNo	▪ Request에 대한 화면 URL를 리턴한다.
getTerminalIpAddress	▪ Request에 대한 클라이언트의 IP 주소 값을 리턴한다.
getTransactionCode	▪ Request에 대한 트랜잭션 코드 값을 리턴한다.
getTransactionHeader	▪ Request에 대한 트랜잭션 헤더 값을 리턴한다.
getTransactionMapId	▪ Request에 대한 트랜잭션 맵 ID를 리턴한다.
getUserHeader	▪ Request에 대한 사용자 헤더를 얻어온다.
getXDataSetOutputString	▪ Response으로 설정된 데이터 셋의 내용을 문자열로 구해온다.

getVersion	<ul style="list-style-type: none"> ▪ XDataSet 라이브러리의 Version 정보를 얻어온다.
isDeleteRecord	<ul style="list-style-type: none"> ▪ 입력용 데이터 셋 레코드가 삭제된 레코드 인지 여부를 반환한다.
isInsertRecord	<ul style="list-style-type: none"> ▪ 입력용 데이터 셋 레코드가 추가된 레코드 인지 여부를 반환한다.
isUpdateRecord	<ul style="list-style-type: none"> ▪ 입력용 데이터 셋 레코드가 변경된 레코드 인지 여부를 반환한다.
returnData	<ul style="list-style-type: none"> ▪ 데이터 셋의 내용을 클라이언트로 송신한다.
returnPartData	<ul style="list-style-type: none"> ▪ 데이터 셋의 현재까지의 데이터를 송신한다 (First-Row 방식)
setCharacterSet	<ul style="list-style-type: none"> ▪ 출력용 데이터에 대한 Character Set을 지정한다.
setColumnNames	<ul style="list-style-type: none"> ▪ 출력용 데이터 셋의 칼럼 이름들을 설정한다.
setData	<ul style="list-style-type: none"> ▪ 출력용 데이터 셋에 문자열 데이터를 설정한다
setDataSetName	<ul style="list-style-type: none"> ▪ 출력용 데이터 셋 이름을 설정한다.
setDataSetNames	<ul style="list-style-type: none"> ▪ 출력용 데이터 셋 이름들을 설정한다.
setDoubleData	<ul style="list-style-type: none"> ▪ 출력용 데이터 셋에 Double형 데이터를 설정한다
setErrorMessage	<ul style="list-style-type: none"> ▪ 출력용 에러 메시지를 설정하는 함수이다.
setFloatData	<ul style="list-style-type: none"> ▪ 출력용 데이터 셋에 Float형 데이터를 설정한다.
setIntData	<ul style="list-style-type: none"> ▪ 출력용 데이터 셋에 Int형 데이터를 설정한다.
setLongData	<ul style="list-style-type: none"> ▪ 출력용 데이터셋에 Long형 데이터를 설정한다.
setMessage	<ul style="list-style-type: none"> ▪ 출력용 안내 메시지를 설정하는 함수이다.
setHttpResponse	<ul style="list-style-type: none"> ▪ Response에 대한 출력 데이터를 송신시 사용할 HttpResponse 오브젝트를 설정
setRecordKey	<ul style="list-style-type: none"> ▪ 출력용 데이터 셋의 Record의 키를 설정한다.
setUserHeader	<ul style="list-style-type: none"> ▪ 출력용 사용자 헤더 문자열을 설정한다.
replaceHTMLTagFilter	<ul style="list-style-type: none"> ▪ 크로스 사이트 스트립팅 처리로 인해 치환된 문자열을 재치환 하여 반환한다.

XDataSet API 설명

getColumnCount

Prototype

```
int getColumnCount(String sXDataSetName);
```

Parameter

Type	Name	설명
String	sXDataSetName	▪ 칼럼 개수를 구할 데이터 셋 이름

Return Value

Type	Name	설명
Int	nColumnCount	▪ 데이터 셋의 칼럼 수

Description

Request 용 데이터 셋의 칼럼 수를 구하는 함수이다.

Sample

```
1 int nColumnCount = xDataSet.getColumnCount("INPUT_DATASET");
```

See Also

[getRecordCount](#), [getColumnNames](#), [getColumnNamesArray](#)

getColumnNames

Prototype

Vector getColumnNames (String sXDataSetName);

Parameter

Type	Name	설명
String	sXDataSetName	▪ 칼럼 이름을 구할 데이터 셋 이름

Return Value

Type	Name	설명
Vector	columnNamesVector	▪ 데이터 셋의 칼럼 이름 정보를 포함하고 있는 Vector 오브젝트

Description

Request 에 대한 데이터 셋의 칼럼 이름들이 포함된 Vector 형 오브젝트를 리턴한다.

Sample

```
1 Vector columnNamesVector = null;
2 columnNamesVector = xDataSet.getColumnNames("INPUT_DATASET");
3 for(int i = 0; i < columnNamesVector.size(); i++) {
4     System.out.println((String)columnNamesVector.get(i));
5 }
```

See Also

getColumnNamesArray

getColumnNamesArray

Prototype

```
String[] getColumnNamesArray (String sXDataSetName);
```

Parameter

Type	Name	설명
String	sXDataSetName	▪ 칼럼 이름을 구할 데이터 셋 이름

Return Value

Type	Name	설명
String[]	columnNamesArray	▪ 데이터 셋의 칼럼 이름 배열

Description

Request 에 대한 데이터 셋의 칼럼 이름을 문자열 배열로 리턴한다.

Sample

```
1 String[] columnNamesArray = null;
2 columnNamesArray = xDataSet.getColumnNamesArray("INPUT_DATASET");
3 for(int i = 0; i < columnNamesArray.length; i++) {
4     System.out.println((String)columnNamesArray[i]);
5 }
```

See Also

[getColumnNames](#)

getData

Prototype

```
String getData(String sDataSetName, String sColumnName, int nRecordIndex);
```

```
String getData(String sDataSetName, int nColumnIndex, int nRecordIndex)
```

Parameter

Type	Name	설명
String	sDataSetName	▪ 데이터를 구할 데이터 셋 이름
String	sColumnName	▪ 데이터를 구할 데이터 셋의 칼럼 이름
int	nRecordIndex	▪ 데이터를 구할 레코드 인덱스
int	nColumnIndex	▪ 데이터를 구할 칼럼 인덱스

Return Value

Type	Name	설명
String	sDataValue	▪ 데이터 셋의 문자열형 데이터

Description

Request 용 데이터 셋의 문자열형 데이터를 얻는 함수이다. 그외의 데이터 형에 대해서는 아래의 함수를 각각의 데이터 유형에 맞추어 호출한다.

- double 형 데이터 : `getDoubleData(String sDataSetName, String sColumnName, int nRecordIndex);`
- float형 데이터 : `getFloatData(String sDataSetName, String sColumnName, int nRecordIndex);`
- int 형 데이터 : `getIntData(String sDataSetName, String sColumnName, int nRecordIndex);`
- long형 데이터 : `getLongData(String sDataSetName, String sColumnName, int nRecordIndex);`

Sample

```

1 String[] columnNamesArray = null;
2 int columnCount = 0;
3 int recordCount = 0;
4 int recordIndex = 0;
4 int columnIndex = 0;
5
```

```
6 // Get Record, Column Count
6 recordCount = xDataSet.getRecordCount("INPUT_DATASET");
7 columnCount = xDataSet.getColumnCount("INPUT_DATASET");
8
9 // Get Column Names Array
10 columnNamesArray = xDataSet.getColumnNamesArray("INPUT_DATASET");
11
12 // Get Data By Column Name
13 for(recordIndex = 0; recordIndex < recordCount; recordIndex++) {
14     for(columnIndex = 0; columnIndex < columnCount; columnIndex++) {
15         System.out.println(xDataSet.getData("INPUT_DATASET",
16             columnNamesArray[columnIndex], recordIndex));
17     }
18 }
19
20 // Get Data By Column Index
21 for(recordIndex = 0; recordIndex < recordCount; recordIndex++) {
22     for(columnIndex = 0; columnIndex < columnCount; columnIndex++) {
23         System.out.println(xDataSet.getData("INPUT_DATASET",
24             columnIndex, recordIndex));
25     }
26 }
```

See Also

[getDoubleData](#), [getFloatData](#), [getIntData](#), [getLongData](#)

getDataSetNames

Prototype

Vector getDataSetNames()

Parameter

Type	Name	설명
		▪

Return Value

Type	Name	설명
Vector	dataSetNamesVector	▪ 데이터 셋의 이름 정보가 들어있는 Vector 오브젝트

Description

Request 및 Response 용 데이터셋들의 이름을 Vector 형으로 리턴한다.

Sample

```
1 Vector dataSetNamesVector = null;
2 dataSetNamesVector = xDataSet.getDataSetNames();
3 for(int i = 0; i < dataSetNamesVector.size(); i++) {
4     System.out.println((String) dataSetNamesVector.get(i));
5 }
```

See Also

getDataSetNamesArray

getDataSetNamesArray

Prototype

```
String[] getDataSetNmesArray()
```

Parameter

Type	Name	설명
		▪

Return Value

Type	Name	설명
String[]	dataSetNamesArray	▪ 데이터 셋의 이름 정보가 들어있는 문자열 배열 오브젝트

Description

Request 및 Response 용 데이터셋들의 이름을 문자형 배열로 리턴한다.

Sample

```
1 String[] dataSetNamesArray = null;
2 dataSetNamesArray = xDataSet.getDataSetNmesArray();
3 for(int i = 0; i < dataSetNamesArray.length; i++) {
4     System.out.println((String) dataSetNamesArray[i]);
5 }
```

See Also

[getDataSetNames](#)

getDataSetType

Prototype

```
String getDataSetType(String sDataSetName);
```

Parameter

Type	Name	설명
String	sDataSetName	<ul style="list-style-type: none"> 데이터를 구할 데이터 셋 이름

Return Value

Type	Name	설명
String	sDataSetType	<ul style="list-style-type: none"> 데이터 셋 타입 XDataSet.XDATA_INOUT : Request/Response용 데이터 셋 XDataSet.XDATA_INPUT : Request용 데이터 셋 XDataSet.XDATA_OUTPUT : Response용 데이터 셋

Description

데이터 셋의 타입을 구하는 함수이다. 데이터 셋의 타입은 Transaction IO Map 에서 입력/출력 부분에 설정된 값에 의해서 결정된다.

Sample

```

1  String[] dataSetNamesArray = null;
2
3  System.out.println(XDataSet.XDATA_OUTPUT);
4  System.out.println(XDataSet.XDATA_INPUT);
5  System.out.println(XDataSet.XDATA_INOUT);
6
7  dataSetNamesArray = xDataSet.getDataSetNamesArray();
8  for(int i = 0; i < dataSetNamesArray.length; i++) {
9      System.out.println(xDataSet.getDataSetType(dataSetNamesArray[i]));
10 }
```

See Also

getDoubleData

Prototype

```
double getDoubleData(String sDataSetName, String sColumnName, int nRecordIndex);
```

```
double getDoubleData(String sDataSetName, int nColumnIndex, int nRecordIndex);
```

Parameter

Type	Name	설명
String	sDataSetName	▪ 데이터를 구할 데이터 셋 이름
String	sColumnName	▪ 데이터를 구할 데이터 셋의 칼럼 이름
int	nRecordIndex	▪ 데이터를 구할 레코드 인덱스

Return Value

Type	Name	설명
double	dDataValue	▪ 데이터 셋의 double형 데이터

Description

Request 용 데이터 셋의 Double 형 데이터를 얻는 함수이다.

Sample

```

1  String[] columnNamesArray = null;
2  int columnCount = 0;
3  int recordCount = 0;
4  int recordIndex = 0;
5  int columnIndex = 0;
6
7  // Get Record, Column Count
8  recordCount = xDataSet.getRecordCount("INPUT_DATASET");
9  columnCount = xDataSet.getColumnCount("INPUT_DATASET");
10
11 // Get Column Names Array
12 columnNamesArray = xDataSet.getColumnNamesArray("INPUT_DATASET");
13
14 // Get Data By Column Name
15 for(recordIndex = 0; recordIndex < recordCount; recordIndex++) {
16     for(columnIndex = 0; columnIndex < columnCount; columnIndex++) {

```



```
17         System.out.println(xDataSet.getDoubleData("INPUT_DATASET",
18             columnNamesArray[columnIndex], recordIndex));
19     }
20 }
```

See Also

[getData](#), [getFloatData](#), [getIntData](#), [getLongData](#)

getFloatData

Prototype

```
float getFloatData(String sDataSetName, String sColumnName, int nRecordIndex);
```

```
float getFloatData(String sDataSetName, int nColumnIndex, int nRecordIndex);
```

Parameter

Type	Name	설명
String	sDataSetName	▪ 데이터를 구할 데이터 셋 이름
String	sColumnName	▪ 데이터를 구할 데이터 셋의 칼럼 이름
int	nRecordIndex	▪ 데이터를 구할 레코드 인덱스

Return Value

Type	Name	설명
float	fDataValue	▪ 데이터 셋의 float형 데이터

Description

Request 용 데이터 셋의 Double 형 데이터를 얻는 함수이다.

Sample

```

1  String[] columnNamesArray = null;
2  int columnCount = 0;
3  int recordCount = 0;
4  int recordIndex = 0;
5  int columnIndex = 0;
6
7  // Get Record, Column Count
8  recordCount = xDataSet.getRecordCount("INPUT_DATASET");
9  columnCount = xDataSet.getColumnCount("INPUT_DATASET");
10
11 // Get Column Names Array
12 columnNamesArray = xDataSet.getColumnNamesArray("INPUT_DATASET");
13
14 // Get Data By Column Name
15 for(recordIndex = 0; recordIndex < recordCount; recordIndex++) {
16     for(columnIndex = 0; columnIndex < columnCount; columnIndex++) {
```

```
17         System.out.println(xDataSet.getFloatData("INPUT_DATASET",
18             columnNamesArray[columnIndex], recordIndex));
19     }
20 }
```

See Also

[getData](#), [getDoubleData](#), [getIntData](#), [getLongData](#)

getFullVersion

Prototype

String getFullVersion()

Parameter

Type	Name	설명
		▪

Return Value

Type	Name	설명
String	sFullVersion	▪ XDataSet 라이브러리의 Full 버전 정보

Description

XDataSet 의 전체 버전 정보를 리턴하는 XDataSet 클래스의 static 함수이다.

Sample

```
1 System.out.println("FULL VER = " + XDataSet.getFullVersion());
```

See Also

getVersion

getIntData

Prototype

```
int getIntData(String sDataSetName, String sColumnName, int nRecordIndex);
```

```
int getIntData(String sDataSetName, int nColumnIndex, int nRecordIndex)
```

Parameter

Type	Name	설명
String	sDataSetName	▪ 데이터를 구할 데이터 셋 이름
String	sColumnName	▪ 데이터를 구할 데이터 셋의 칼럼 이름
int	nRecordIndex	▪ 데이터를 구할 레코드 인덱스

Return Value

Type	Name	설명
int	nDataValue	▪ 데이터 셋의 int형 데이터

Description

Request 용 데이터 셋의 Double 형 데이터를 얻는 함수이다.

Sample

```

1  String[] columnNamesArray = null;
2  int columnCount = 0;
3  int recordCount = 0;
4  int recordIndex = 0;
5  int columnIndex = 0;
6
7  // Get Record, Column Count
8  recordCount = xDataSet.getRecordCount("INPUT_DATASET");
9  columnCount = xDataSet.getColumnCount("INPUT_DATASET");
10
11 // Get Column Names Array
12 columnNamesArray = xDataSet.getColumnNamesArray("INPUT_DATASET");
13
14 // Get Data By Column Name
15 for(recordIndex = 0; recordIndex < recordCount; recordIndex++) {
16     for(columnIndex = 0; columnIndex < columnCount; columnIndex++) {
```

```
17         System.out.println(xDataSet.getIntData("INPUT_DATASET",
18             columnNamesArray[columnIndex], recordIndex));
19     }
20 }
```

See Also

[getData](#), [getDoubleData](#), [getFloatData](#), [getLongData](#)

getLongData

Prototype

```
long getLongData(String sDataSetName, String sColumnName, int nRecordIndex);
```

```
long getLongData(String sDataSetName, int nColumnIndex, int nRecordIndex)
```

Parameter

Type	Name	설명
String	sDataSetName	▪ 데이터를 구할 데이터 셋 이름
String	sColumnName	▪ 데이터를 구할 데이터 셋의 칼럼 이름
int	nRecordIndex	▪ 데이터를 구할 레코드 인덱스

Return Value

Type	Name	설명
long	IDataValue	▪ 데이터 셋의 long형 데이터

Description

Request 용 데이터 셋의 Double 형 데이터를 얻는 함수이다.

Sample

```

1  String[] columnNamesArray = null;
2  int columnCount = 0;
3  int recordCount = 0;
4  int recordIndex = 0;
5  int columnIndex = 0;
6
7  // Get Record, Column Count
8  recordCount = xDataSet.getRecordCount("INPUT_DATASET");
9  columnCount = xDataSet.getColumnCount("INPUT_DATASET");
10
11 // Get Column Names Array
12 columnNamesArray = xDataSet.getColumnNamesArray("INPUT_DATASET");
13
14 // Get Data By Column Name
15 for(recordIndex = 0; recordIndex < recordCount; recordIndex++) {
16     for(columnIndex = 0; columnIndex < columnCount; columnIndex++) {
```

```
17         System.out.println(xDataSet.getIntData("INPUT_DATASET",
18             columnNamesArray[columnIndex], recordIndex));
19     }
20 }
```

See Also

[getData](#), [getDoubleData](#), [getFloatData](#), [getIntData](#)

getOutputColumnNames

Prototype

Vector getOutputColumnNames (String sXDataSetName);

Parameter

Type	Name	설명
String	sXDataSetName	▪ 칼럼 이름을 구할 데이터 셋 이름

Return Value

Type	Name	설명
Vector	columnNamesVector	▪ 데이터 셋의 칼럼 이름 정보를 포함하고 있는 Vector 오브젝트

Description

Response 용 데이터 셋의 칼럼 이름들이 포함된 Vector 형 오브젝트를 리턴한다.

Sample

```
1 Vector columnNamesVector = null;
2 columnNamesVector = xDataSet.getColumnNames("OUTPUT_DATASET");
3 for(int i = 0; i < columnNamesVector.size(); i++) {
4     System.out.println((String)columnNamesVector.get(i));
5 }
```

See Also

getOutputColumnNamesArray

getOutputColumnNamesArray

Prototype

```
String[] getOutputColumnNamesArray (String sXDataSetName);
```

Parameter

Type	Name	설명
String	sXDataSetName	<ul style="list-style-type: none"> 칼럼 이름을 구할 데이터 셋 이름

Return Value

Type	Name	설명
String[]	columnNamesArray	<ul style="list-style-type: none"> 데이터 셋의 칼럼 이름 배열

Description

Response 용 데이터 셋의 칼럼 이름을 문자열 배열로 리턴한다.

Sample

```

1 String[] columnNamesArray = null;
2 columnNamesArray = xDataSet.getOutputColumnNamesArray("INPUT_DATASET");
3 for(int i = 0; i < columnNamesArray.length; i++) {
4     System.out.println((String)columnNamesArray[i]);
5 }
```

See Also

getOutputColumnNames

getOutputData

Prototype

```
String getOutputData(String sDataSetName, String sColumnName, int nRecordIndex);
```

```
String getOutputData(String sDataSetName, int nColumnIndex, int nRecordIndex)
```

Parameter

Type	Name	설명
String	sDataSetName	▪ 데이터를 구할 데이터 셋 이름
String	sColumnName	▪ 데이터를 구할 데이터 셋의 칼럼 이름
int	nRecordIndex	▪ 데이터를 구할 레코드 인덱스
int	nColumnIndex	▪ 데이터를 구할 칼럼 인덱스

Return Value

Type	Name	설명
String	sDataValue	▪ 데이터 셋의 문자열형 데이터

Description

Response 용 데이터 셋에 설정된 문자열형 데이터를 얻는 함수이다.

Sample

1	
---	--

See Also

getOutputColumnCount, getOutputRecordCount

getOutputRecordCount

Prototype

```
int getOutputRecordCount(String sDataSetName)
```

Parameter

Type	Name	설명
String	sDataSetName	▪ 레코드 수를 구할 데이터 셋 이름

Return Value

Type	Name	설명
int	recordCount	▪ 데이터 셋의 레코드 수

Description

Response 용 데이터 셋의 레코드 수를 리턴한다.

Sample

```
1 int recordCount = 0;
2 recordCount = xDataSet.getOutputRecordCount("OUTPUT_DATASET");
3 System.out.println("Record Count = " + recordCount);
```

See Also

[getOutputData](#)

getRecordCount

Prototype

```
int getRecordCount(String sDataSetName)
```

Parameter

Type	Name	설명
String	sDataSetName	▪ 레코드 수를 구할 데이터 셋 이름

Return Value

Type	Name	설명
int	recordCount	▪ 데이터 셋의 레코드 수

Description

Request 에 대한 데이터 셋의 레코드 수를 리턴한다.

Sample

```
1 int recordCount = 0;
2 recordCount = xDataSet.getRecordCount("INPUT_DATASET");
3 System.out.println("Record Count = " + recordCount);
```

See Also

getRecordStatus, getRecordKey

getRecordKey

Prototype

String getRecordKey(String sDataSetName, int nRecordIndex)

Parameter

Type	Name	설명
String	sDataSetName	▪ 데이터를 구할 데이터 셋 이름
Int	nRecordIndex	▪ 데이터를 구할 레코드 인덱스

Return Value

Type	Name	설명
String	sRecordKey	▪ 레코드의 키 값

Description

Request 용 데이터 셋의 레코드 키 값을 얻는 함수이다. 레코드 키 값은 Viewer 에서 내부적으로 관리하는 레코드의 대한 유일한 키값이다. 이 값을 사용하여 Response 데이터 셋의 데이터 설정시 특정 레코드의 값을 업데이트 할 수 있다.

Sample

```

1  int recordCount = 0;
2  int recordIndex = 0;
3
4  recordCount = xDataSet.getRecordCount("INPUT_DATASET");
5  System.out.println("Record Count = " + recordCount);
6
7  for(recordIndex = 0; recordIndex < recordCount; recordIndex++) {
8      System.out.println("KEY = " +
9          xDataSet.getRecordKey("INPUT_DATASET", recordIndex));
10 }
```

See Also

setRecordKey

getRecordType

Prototype

String getRecordType(String sDataSetName, int nRecordIndex)

Parameter

Type	Name	설명
String	sDataSetName	<ul style="list-style-type: none"> 데이터를 구할 데이터 셋 이름
Int	nRecordIndex	<ul style="list-style-type: none"> 데이터를 구할 레코드 인덱스

Return Value

Type	Name	설명
String	sRecordType	<ul style="list-style-type: none"> 레코드의 상태 값 XDataSet.XDATA_DELETE : 삭제된 레코드 XDataSet.XDATA_INSERT : 추가된 레코드 XDataSet.XDATA_UPDATE : 수정된 레코드 XDataSet.XDATA_NONE : 변경되지 않은 레코드

Description

Request 용 데이터 셋의 레코드의 상태 값을 얻는 함수이다. 레코드 상태값은 Viewer 에서 내부적으로 관리하는 레코드의 대한 상태 정보이다.

Sample

```

1   int recordCount = 0;
2   int recordIndex = 0;
3
4   System.out.println(XDataSet.XDATA_DELETE);
5   System.out.println(XDataSet.XDATA_INSERT);
6   System.out.println(XDataSet.XDATA_UPDATE);
7   System.out.println(XDataSet.XDATA_NONE);
8
9   recordCount = xDataSet.getRecordCount("INPUT_DATASET");
10  System.out.println("Record Count = " + recordCount);
11
12  for(recordIndex = 0; recordIndex < recordCount; recordIndex++) {

```

```
13     System.out.println("TYPE = " +  
14         xDataSet.getRecordType("INPUT_DATASET", recordIndex));  
15 }
```

See Also

isDeleteRecord, isInsertRecord, isUpdateRecord

getReturnByteArrayData

Prototype

byte[] getReturnByteArrayData (String sCharSet)

Parameter

Type	Name	설명
String	sCharSet	▪ 문자열 셋

Return Value

Type	Name	설명
Byte[]	baReturnData	▪ Reponse용 바이트 어레이형 데이터

Description

Response 용 데이터를 바이트 배열형으로 리턴한다.

Sample

```
1 byte[] returnData = xDataSet.getReturnByteArrayData ("UTF-8");
```

See Also

getReturnStringData

getReturnStringData

Prototype

String getReturnStringData ()

Parameter

Type	Name	설명
		▪

Return Value

Type	Name	설명
String	sReturnData	▪ Reponse용 문자열형 데이터

Description

Response 용 데이터를 문자열형으로 리턴한다.

Sample

```
1 String returnData = xDataSet.getReturnStringData();
```

See Also

[getTerminalIpAddress](#), [getTransactionCode](#), [getTransactionHeader](#), [getTransactionMapId](#)

getScreenNo

Prototype

String getScreenNo()

Parameter

Type	Name	설명
		▪

Return Value

Type	Name	설명
String	sScreenUrl	▪ 화면 URL

Description

Request 한 Viewer 단의 화면 URL 을 리턴한다. 화면 URL 은 DevStudio 내에서 화면이 저장된 경로를 의미하며, 만약 0010103 화면라는 화면이 PSB→TLR 디렉토리 경로로 저장되어 있는 경우, "/PSB/TLR/0010103"가 리턴된다. 이 정보는 Viewer 에서 requestsubmit 함수 호출시 Viewer 내부에서 자동으로 설정되어 서버로 송신된다.

Sample

```
1 System.out.println("SCREEN NO = " + xDataSet.getScreenNo());
```

See Also

getTerminalIpAddress, getTransactionCode, getTransactionHeader, getTransactionMapId

getTerminalIpAddress

Prototype

String getTerminalIpAddress()

Parameter

Type	Name	설명
		▪

Return Value

Type	Name	설명
String	slpAddr	▪ 사용자 PC의 IP 주소

Description

Request 한 사용자 PC의 IP 주소를 리턴한다. 이 정보는 Viewer 에서 requestsubmit 함수 호출시 Viewer 내부에서 자동으로 설정되어 서버로 송신된다.

Sample

```
1 System.out.println("IP ADDRESS = " + xDataSet.getTerminalIpAddress());
```

See Also

getScreenNo, getTransactionCode, getTransactionHeader, getTransactionMapId

getTransactionCode

Prototype

```
String getTransactionCode()
```

Parameter

Type	Name	설명
		▪

Return Value

Type	Name	설명
String	sTransactionCode	▪ Request를 요청한 화면의 "trancode" 속성 값

Description

Request 를 요청한 화면의 "trancode" 값을 리턴한다. 화면의 "trancode"와 "tranheader" 속성은 프로젝트 프로젝트 진행시 임의로 지정할 수 있는 값이다. Viewer 엔진에서는 이 값을 사용하지 않는다. 이 정보는 Viewer 에서 requestsubmit 함수 호출시 Viewer 내부에서 자동으로 설정되어 서버로 송신된다.

Sample

```
1 System.out.println("TR CODE = " + xDataSet.getTransactionCode());
```

See Also

getScreenNo, getTerminalIpAddress, getTransactionHeader, getTransactionMapId

getTransactionHeader

Prototype

String getTransactionHeader()

Parameter

Type	Name	설명
		▪

Return Value

Type	Name	설명
String	sTransactionHeader	▪ Request를 요청한 화면의 "tranheader" 값

Description

Request 를 요청한 화면의 "tranheader" 값을 리턴한다. 화면의 "tranheader" 속성은 프로젝트 프로젝트 진행시 임의로 지정할 수 있는 값이다. Viewer 엔진에서는 이 값을 사용하지 않는다. 이 정보는 Viewer 에서 requestsubmit 함수 호출시 Viewer 내부에서 자동으로 설정되어 서버로 송신된다.

Sample

```
1 System.out.println("TR HEADER = " + xDataSet.getTransactionHeader());
```

See Also

getScreenNo, getTerminalIpAddress, getTransactionCode, getTransactionMapId

getTransactionMapId

Prototype

String getTransactionMapId()

Parameter

Type	Name	설명
		▪

Return Value

Type	Name	설명
String	sTransactionMapId	▪ Request를 요청한 화면의 Transaction Map ID 값

Description

Request 를 요청한 화면의 Transaction Map 의 ID 값을 리턴한다. 화면의 Transaction Map 의 ID 값은 Viewer 의 screen 오브젝트의 requestsubmit() 함수의 첫번째 파라미터로 전달된 값이다. 이 정보는 Viewer 에서 requestsubmit 함수 호출시 Viewer 내부에서 자동으로 설정되어 서버로 송신된다.

Sample

```
1 System.out.println("MAP ID = " + xDataSet.getTransactionMapId());
```

See Also

getScreenNo, getTerminalIpAddress, getTransactionCode, getTransactionHeader

getUserHeader

Prototype

String getUserHeader()

Parameter

Type	Name	설명
		▪

Return Value

Type	Name	설명
String	sUserHeader	▪ Request를 요청한 화면의 사용자 헤더 값

Description

Request 를 요청한 화면에서 설정한 사용자 헤더 정보의 값을 리턴한다. 사용자 헤더값은 requestsubmit() 함수 호출 이전에 screen.setUserheader() 함수를 통해서 설정된 값이다. 사용자 헤더는 데이터 셋 이외에 추가적인 정보를 서버로 송신하는 용도로 사용된다.

Sample

```
1 System.out.println("USER HEADER = " + xDataSet.getUserHeader());
```

See Also

setUserHeader

getXDataSetOutputDataString

Prototype

```
String getXDataSetOutputDataString ()
```

Parameter

Type	Name	설명
		▪

Return Value

Type	Name	설명
String	sOutputString	▪ Response용 데이터 셋에 들어있는 데이터에 대한 디버깅용 문자열

Description

Response 용 데이터 셋에 설정된 데이터에 대한 정보를 문자열로 리턴한다. Response 용 데이터 셋에 설정되어 있는 값을 디버깅 용도로 확인하기 위해서 사용된다.

Sample

```
1 System.out.println("OUTPUT = " + xDataSet.getXDataSetOutputDataString());
```

See Also

getVersion

Prototype

String getVersion()

Parameter

Type	Name	설명
		▪

Return Value

Type	Name	설명
String	sVersion	▪ XDataSet 라이브러리의 버전

Description

XDataSet 의 전체 버전 정보를 리턴하는 XDataSet 클래스의 static 함수이다.

Sample

```
1 System.out.println("VER = " + XDataSet.getVersion());
```

See Also

getFullVersion

isDeleteRecord

Prototype

```
bool isDeleteRecord(String sDataSetName, int nRecordIndex)
```

Parameter

Type	Name	설명
String	sDataSetName	▪ 데이터를 구할 데이터 셋 이름
int	nRecordIndex	▪ 데이터를 구할 레코드 인덱스

Return Value

Type	Name	설명
boolean	true/false	▪ 삭제된 레코드 여부

Description

Request 용 데이터 셋의 레코드가 삭제된 레코드인지를 판단한다.

Sample

```

1  int recordCount = 0;
2  int recordIndex = 0;
3
4  recordCount = xDataSet.getRecordCount("INPUT_DATASET");
5  System.out.println("Record Count = " + recordCount);
6
7  for(recordIndex = 0; recordIndex < recordCount; recordIndex++) {
8      if(xDataSet.isDeleteRecord("INPUT_DATASET", recordIndex)) {
9          System.out.println("DELETE RECORD");
10     }
11 }
```

See Also

isInsertRecord, isUpdateRecord

isInsertRecord

Prototype

```
bool isInsertRecord(String sDataSetName, int nRecordIndex)
```

Parameter

Type	Name	설명
String	sDataSetName	▪ 데이터를 구할 데이터 셋 이름
int	nRecordIndex	▪ 데이터를 구할 레코드 인덱스

Return Value

Type	Name	설명
boolean	true/false	▪ 추가된 레코드 여부

Description

Request 용 데이터 셋의 레코드가 추가된 레코드인지를 판단한다.

Sample

```

1  int recordCount = 0;
2  int recordIndex = 0;
3
4  recordCount = xDataSet.getRecordCount("INPUT_DATASET");
5  System.out.println("Record Count = " + recordCount);
6
7  for(recordIndex = 0; recordIndex < recordCount; recordIndex++) {
8      if(xDataSet.isInsertRecord("INPUT_DATASET", recordIndex)) {
9          System.out.println("INSERT RECORD");
10     }
11 }
```

See Also

isDeleteRecord, IsUpdateRecord

isUpdateRecord

Prototype

```
boolean isUpdateRecord(String sDataSetName, int nRecordIndex)
```

Parameter

Type	Name	설명
String	sDataSetName	▪ 데이터를 구할 데이터 셋 이름
Int	nRecordIndex	▪ 데이터를 구할 레코드 인덱스

Return Value

Type	Name	설명
boolean	true/false	▪ 변경된 레코드 여부

Description

Request 용 데이터 셋의 레코드가 변경된 레코드인지를 판단한다.

Sample

```

1  int recordCount = 0;
2  int recordIndex = 0;
3
4  recordCount = xDataSet.getRecordCount("INPUT_DATASET");
5  System.out.println("Record Count = " + recordCount);
6
7  for(recordIndex = 0; recordIndex < recordCount; recordIndex++) {
8      if(xDataSet.isInsertRecord("INPUT_DATASET", recordIndex)) {
9          System.out.println("UPDATE RECORD");
10     }
11 }
```

See Also

isDeleteRecord, issInsertRecord

returnData

Prototype

```
void returnData()
```

Parameter

Type	Name	설명
		▪

Return Value

Type	Name	설명
		▪

Description

Response 용 데이터 셋의 데이터를 Viewer 단으로 전송한다. 이 함수가 호출된 이후에 추가적인 데이터를 Viewer 쪽으로 전송할 수 없다.

Sample

```

1  int recordIndex = 0;
2  int columnIndex = 0;
3  int recordCount = 10;
4  int columnCount = 5;
5
6  for(recordIndex = 0; recordIndex < recordCount; recordIndex++) {
7      for(columnIndex = 0; columnIndex < columnCount; columnIndex++) {
8          xDataSet.setData(
9              "OUTPUT_DATASET", "COL_" + columnIndex, recordIndex,
10             "DATA_" + recordIndex + "_" + columnIndex);
11     }
12 }
13
14 xDataSet.returnData();
```

See Also

[returnPartData](#)

returnPartData

Prototype

```
void returnPartData()
```

Parameter

Type	Name	설명
		▪

Return Value

Type	Name	설명
		▪

Description

함수 호출 시점까지의 Response 용 데이터 셋의 데이터를 Viewer 쪽으로 전송한다. 이 함수를 데이터셋의 레코드 단위로 데이터가 채워진 시점에 호출할 수 있다. returnData 함수를 호출하기 전에 여러 번 호출하여도 무관한다. returnData 함수가 호출된 이후에는 returnPartData 함수를 호출할 수 없다.

Sample

```

1  int recordIndex = 0;
2  int columnIndex = 0;
3  int recordCount = 10;
4  int columnCount = 5;
5
6  for(recordIndex = 0; recordIndex < recordCount; recordIndex++) {
7      for(columnIndex = 0; columnIndex < columnCount; columnIndex++) {
8          xDataSet.setData(
9              "OUTPUT_DATASET", "COL_" + columnIndex, recordIndex,
10             "DATA_" + recordIndex + "_" + columnIndex);
11         }
12
13         // Return Part Data Per 10 Records
14         if((recordIndex % 10) == 0) {
15             xDataSet.returnPartData();
16             Thread.sleep(2000);    // For Test, Sleep For 2 Seconds
17         }

```



```
}  
  
xDataSet.returnData();
```

See Also

[returnData](#)

setCharacterSet

Prototype

```
void setCharacterSet(String sCharacterSet)
```

Parameter

Type	Name	설명
String	sCharacterSet	출력 데이터로 사용할 Character Set 문자열

Return Value

Type	Name	설명

Description

Response 용 데이터 셋의 데이터를 Viewer 단으로 전송하기 위해 사용하는 Character Set 을 지정한다. 서버 비즈니스 플랫폼의 기본 Character Set 이 "EUC-KR"이 아닌 경우에, 해당 Character Set 을 지정한다.

Sample

```

1  int recordIndex = 0;
2  int columnIndex = 0;
3  int recordCount = 10;
4  int columnCount = 5;
5
6  for(recordIndex = 0; recordIndex < recordCount; recordIndex++) {
7      for(columnIndex = 0; columnIndex < columnCount; columnIndex++) {
8          xDataSet.setData(
9              "OUTPUT_DATASET", "COL_" + columnIndex, recordIndex,
10             "HANGUL_DATA_" + recordIndex + "_" + columnIndex);
11         }
12     }
13
14     xDataSet.setCharacterSet("UTF-8");
15
16     xDataSet.returnData();

```

See Also

`returnData`, `returnPartData`

setColumnNames

Prototype

```
void setColumnNames(string sDataSetName, String[] saColumnNames)
```

```
void setColumnNames(string sDataSetName, Vector vColumnNames)
```

Parameter

Type	Name	설명
String	sDataSetName	▪ Resposne용 데이터 셋 이름
String[] 또는 Vector	saColumnNames 또는 vColumnNames	▪ Resposne용 데이터 셋의 칼럼 이름 배열 또는 Vector

Return Value

Type	Name	설명
		▪

Description

XDataSet 오브젝트를 Response 전용으로 생성한 경우에, Response 데이터를 설정하기 전에 Response 용 데이터 셋의 칼럼 이름을 설정하는 함수이다. XDataSet 오브젝트를 Response 전용으로 생성한 경우에, setColumnNames() 함수를 호출하기 전에, 반드시 해당 칼럼 정보를 지정할 데이터 셋의 이름이, setDataSetName() 또는 setDataSetNames() 함수를 통해서 설정이 되어 있어야 한다. HttpRequest, HttpResponse 오브젝트를 파라미터로 받아서 생성한 경우에는 호출하면 안된다.

Sample

```

1   int recordIndex = 0;
2   int columnIndex = 0;
3   int recordCount = 20;
4   int columnCount = 0;
5
6   // Create New XDataSet Object For Output
7   XDataSet xDataSet = new XDataSet();
8
9   // Set HTTP Response Object

```

```
10 xDataSet.setHttpResponse(httpRsp);
11
12 String[] saColumnNames = new String[2];
13 saColumnNames[0] = "OUTPUT_COLUMN_1";
14 saColumnNames[1] = "OUTPUT_COLUMN_2";
15 columnCount = saColumnNames.length;
16
17 xDataSet.setDataSetName("OUTPUT_DATASET");
18 xDataSet.setColumnNames("OUTPUT_DATASET", saColumnNames);
19
20 for(recordIndex = 0; recordIndex < recordCount; recordIndex++) {
21     for(columnIndex = 0; columnIndex < columnCount; columnIndex++) {
22         xDataSet.setData(
23             "OUTPUT_DATASET",
24             saColumnNames[columnIndex],
25             recordIndex,
26             "DATA_" + recordIndex + "_" + columnIndex);
27     }
28 }
29
30 xDataSet.returnData();
```

See Also

SetDataSetName, SetDataSetNames

setData

Prototype

```
void setData(string sDataSetName, String sColumnName, int nRecordIndex, String sValue);
```

```
void setData(string sDataSetName, int nColumnIndex, int nRecordIndex, String sValue)
```

Parameter

Type	Name	설명
String	sDataSetName	▪ Resposne용 데이터 셋 이름
String	sColumnName	▪ Resposne용 데이터 셋의 칼럼 이름
int	nRecordIndex	▪ 데이터 셋 레코드 인덱스
String	sValue	▪ 설정할 데이터

Return Value

Type	Name	설명
		▪

Description

Response 용 데이터를 설정하는 함수이다. 설정된 데이터는 ReturnPartData 또는 ReturnData 함수 호출시 Viewer 에게 전송된다. xDataSet 오브젝트가 Response 전용으로 생성된 경우에는 setData 함수를 호출하기 전에 setDataSetName, setColumnNames 함수를 호출하여 Response 용 데이터 셋의 구조를 설정해야 한다.

Sample

```

1 String valueString = 3;
2
3 xDataSet.setData("OUTPUT_DATASET", "COLUMN_STRING",
4                 recordIndex, valueString);
5
6 xDataSet.returnData();
```

See Also

setDoubleData, setIntData, setFloatData, setLongData

setDataSetName

Prototype

```
void setDataSetName(string sDataSetName)
```

Parameter

Type	Name	설명
String	sDataSetName	▪ Resposne용 데이터 셋 이름

Return Value

Type	Name	설명
		▪

Description

Response 용 데이터 셋의 이름을 추가하는 함수이다. XDataSet 오브젝트를 Response 전용으로 생성한 경우에 Response 데이터를 설정하기 전에 반드시 호출돼야 한다. HttpRequest, HttpResponse 오브젝트를 파라미터로 받아서 생성한 경우에는 호출하면 안된다.

Sample

```
1 xDataSet.SetDataSetName("OUTPUT_DATASET");
```

See Also

SetDataSetNames, SetColumnNames

setDataSetNames

Prototype

```
void setDataSetNames(String[] saDataSetNames)
```

```
void setDataSetNames(Vector vDataSetNames)
```

Parameter

Type	Name	설명
String[] 또는 Vector	saDataSetNames 또는 vDataSetNames	<ul style="list-style-type: none"> Response용 데이터 셋 이름 배열 또는 Vector

Return Value

Type	Name	설명
		<ul style="list-style-type: none">

Description

Response 용 데이터 셋의 이름을 추가하는 함수이다. XDataSet 오브젝트를 Response 전용으로 생성한 경우에 Response 데이터를 설정하기 전에 반드시 호출돼야 한다. HttpRequest, HttpResponse 오브젝트를 파라미터로 받아서 생성한 경우에는 호출하면 안된다.

Sample

```

1 String[] saDataSetNames = new String[2];
2
3 saDataSetNames[0] = "OUTPUT_DATASET_1";
4 saDataSetNames[1] = "OUTPUT_DATASET_2";
5
6 xDataSet.SetDataSetNames(saDataSetNames);

```

See Also

SetDataSetName, SetColumnNames

setDoubleData

Prototype

```
void setDoubleData(string sDataSetName, String sColumnName, int nRecordIndex, double dValue);
```

```
void setDoubleData(string sDataSetName, int nColumnIndex, int nRecordIndex, double dValue)
```

Parameter

Type	Name	설명
String	sDataSetName	▪ Resposne용 데이터 셋 이름
String	sColumnName	▪ Resposne용 데이터 셋의 칼럼 이름
int	nRecordIndex	▪ 데이터 셋 레코드 인덱스
double	dValue	▪ 설정할 데이터

Return Value

Type	Name	설명
		▪

Description

Response 용 데이터를 설정하는 함수이다. 설정된 데이터는 ReturnPartData 또는 ReturnData 함수 호출시 Viewer 에게 전송된다. xDataSet 오브젝트가 Response 전용으로 생성된 경우에는 SetData 함수를 호출하기 전에 SetDataSetName, SetColumnNames 함수를 호출하여 Response 용 데이터 셋의 구조를 설정해야 한다.

Sample

```

1 double valueDouble = 3;
2
3 xDataSet.setFloatData("OUTPUT_DATASET", "COLUMN_DOUBLE",
4                       recordIndex, valueDouble);
5
6 xDataSet.returnData();
```

See Also

setData, setIntData, setFloatData, setLongData

setErrorMessage

Prototype

```
void setErrorMessage(String sErrCode, String sMessage);
```

Parameter

Type	Name	설명
String	sErrCode	▪ 에러 메시지 코드
String	sMessage	▪ 에러 메시지

Return Value

Type	Name	설명
		▪

Description

Request 처리중 오류가 발생한 경우에, Viewer 쪽으로 전송할 에러 메시지 코드와 에러 메시지를 설정하는 함수이다. 이 함수가 호출된 이후에 ReturnData 함수를 호출한 경우에, 화면의 on_submit_complete 콜백함수의 recv_code 와 recv_msg 파라미터에 설정되고, result 항목에는 "0"으로 설정된다. sErrCode 와 sMessage 파라미터는 개발자가 자유롭게 설정이 가능하다.

Sample

```

1  string sErrCode = "ER0001";
2  string sMessage = "Invalid Data Format";
3
4  xDataSet.SetErrorMessage(sErrCode, sMessage);
5  xDataSet.ReturnData();

```

See Also

setMessage, returnData

setFloatData

Prototype

```
void setFloatData(String sDataSetName, String sColumnName, int nRecordIndex, float fValue);
```

```
void setFloatData(String sDataSetName, int nColumnIndex, int nRecordIndex, float fValue)
```

Parameter

Type	Name	설명
String	sDataSetName	▪ Resposne용 데이터 셋 이름
String	sColumnName	▪ Resposne용 데이터 셋의 칼럼 이름
int	nRecordIndex	▪ 데이터 셋 레코드 인덱스
float	fValue	▪ 설정할 데이터

Return Value

Type	Name	설명
		▪

Description

Response 용 데이터를 설정하는 함수이다. 설정된 데이터는 ReturnPartData 또는 ReturnData 함수 호출시 Viewer 에게 전송된다. xDataSet 오브젝트가 Response 전용으로 생성된 경우에는 SetData 함수를 호출하기 전에 SetDataSetName, SetColumnNames 함수를 호출하여 Response 용 데이터 셋의 구조를 설정해야 한다.

Sample

```

1 float valueFloat = 3;
2
3 xDataSet.setFloatData("OUTPUT_DATASET", "COLUMN_FLOAT",
4                       recordIndex, valueFloat);
5
6 xDataSet.returnData();
```

See Also

setData, setDoubleData, setIntData, setLongData

setIntData

Prototype

```
void setIntData(String sDataSetName, String sColumnName, int nRecordIndex, int nValue);
```

```
void setIntData(String sDataSetName, int nColumnIndex, int nRecordIndex, int nValue)
```

Parameter

Type	Name	설명
String	sDataSetName	▪ Resposne용 데이터 셋 이름
String	sColumnName	▪ Resposne용 데이터 셋의 칼럼 이름
int	nRecordIndex	▪ 데이터 셋 레코드 인덱스
Int	nValue	▪ 설정할 데이터

Return Value

Type	Name	설명
		▪

Description

Response 용 데이터를 설정하는 함수이다. 설정된 데이터는 ReturnPartData 또는 ReturnData 함수 호출시 Viewer 에게 전송된다. xDataSet 오브젝트가 Response 전용으로 생성된 경우에는 setData 함수를 호출하기 전에 setDataSetName, setColumnNames 함수를 호출하여 Response 용 데이터 셋의 구조를 설정해야 한다.

Sample

```

1  int valueInt = 3;
2
3  xDataSet.setFloatData("OUTPUT_DATASET", "COLUMN_DOUBLE",
4                          recordIndex, valueInt);
5
6  xDataSet.returnData();

```

See Also

setData, setDoubleData, setFloatData, setLongData

setLongData

Prototype

```
void setLongData(String sDataSetName, String columnName, int nRecordIndex, long lValue);
```

```
void setLongData(String sDataSetName, int nColumnIndex, int nRecordIndex, long lValue)
```

Parameter

Type	Name	설명
String	sDataSetName	▪ Resposne용 데이터 셋 이름
String	sColumnName	▪ Resposne용 데이터 셋의 칼럼 이름
int	nRecordIndex	▪ 데이터 셋 레코드 인덱스
long	lValue	▪ 설정할 데이터

Return Value

Type	Name	설명
		▪

Description

Response 용 데이터를 설정하는 함수이다. 설정된 데이터는 ReturnPartData 또는 ReturnData 함수 호출시 Viewer 에게 전송된다. xDataSet 오브젝트가 Response 전용으로 생성된 경우에는 SetData 함수를 호출하기 전에 SetDataSetName, SetColumnNames 함수를 호출하여 Response 용 데이터 셋의 구조를 설정해야 한다.

Sample

```

1  long valueLong = 3;
2
3  xDataSet.setFloatData("OUTPUT_DATASET", "COLUMN_LONG",
4                          recordIndex, valueLong);
5
6  xDataSet.returnData();
```

See Also

setData, setDoubleData, setIntData, setFloatData

setMessage

Prototype

```
void setMessage(string sMsgCode, string sMessage);
```

Parameter

Type	Name	설명
String	sMsgCode	▪ 에러 메시지 코드
String	sMessage	▪ 에러 메시지

Return Value

Type	Name	설명
		▪

Description

Request 가 정상적으로 처리되었으나, Viewer 에게 추가적인 메시지를 전달하기 위해서 사용하는 함수이다. 이 함수가 호출된 이후에 ReturnData 함수를 호출한 경우에, 화면의 on_submit_complete 콜백함수의 recv_code 와 recv_msg 파라미터에 설정된다.

Sample

```

1 String sMsgCode = "INFO_0001";
2 String sMessage = "Success";
3
4 xDataSet.SetErrorMessage(sMsgCode, sMessage);
5
6 xDataSet.ReturnData();
```

See Also

setMessage, setUserHeader, returnData

setHttpResponse

Prototype

```
void setHttpResponse(HttpResponse oHttpRsp)
```

Parameter

Type	Name	설명
HttpResponse	oHttpRsp	<ul style="list-style-type: none"> HttpResponse 오브젝트

Return Value

Type	Name	설명
		<ul style="list-style-type: none">

Description

Response 용 데이터 셋의 데이터를 Viewer 에게 전송하기 위해서 사용할 HttpResponse 오브젝트를 설정하는 함수이다. XDataSet 오브젝트를 Response 전용으로 생성한 경우에 Response 데이터를 설정하기 전에 반드시 호출되어야 한다. HttpRequest, HttpResponse 오브젝트를 파라미터로 받아서 생성한 경우에는 호출할 필요가 없다.

Sample

```
1 xDataSet.setHttpResponse ( httpResponse ) ;
```

See Also

returnData, returnPartData

setRecordKey

Prototype

```
void setRecordKey(string sDataSetName, int nRecordIndex, String sRecordKey)
```

Parameter

Type	Name	설명
String	sDataSetName	▪ Resposne용 데이터 셋 이름
Int	nRecordIndex	▪ 데이터 셋의 레코드 인덱스
String	sRecordKey	▪ 설정할 데이터 셋의 레코드 키 값

Return Value

Type	Name	설명
		▪

Description

Response 용 데이터 셋의 특정 레코드의 레코드 키 값을 설정하는 함수이다. 트랜잭션 I/O 정의시 출력 부분의 유형을 "UPDATE"로 지정한 경우, 해당 키 값을 이용해서 Viewer 가 기존에 가지고 있는 데이터를 갱신한다. 설정할 레코드 키 값은 GetRecordKey 함수를 이용해서 사용한다. 트랜잭션 I/O 정의시 출력 부분이 "CLEAR" 방식인 경우에는 아무런 의미가 없다. Viewer 단에서 RecordKey 값을 설정된 데이터를 수신한 경우에, 기존 데이터에서 해당 레코드 키에 매칭되는 레코드가 없을 경우에는, 기존 데이터에 추가된다.

Sample

```

1  int    nRecordIndex = 0;
2  String sRecordKey:
3
4  sRecordKey = xDataSet.getRecordKey("INPUT_DATASET", nRecordIndex);
5  xDataSet.setData("OUTPUT_DATASET", "COLUMN_UPDATE",
6                    nRecordIndex, "UPDATE_DATA");
7  xDataSet.setRecordKey("OUTPUT_DATASET", nRecordIndex, sRecordKey);

```

See Also

getRecordKey

setUserHeader

Prototype

```
void setUserHeader(String sUserHeader);
```

Parameter

Type	Name	설명
String	sUserHeader	▪ 설정할 데이터 셋의 레코드 키 값

Return Value

Type	Name	설명
		▪

Description

Response 용 데이터 셋과 함께, Viewer 쪽으로 송신될 사용자 헤더를 설정하는 함수이다. 화면의 on_submit_complete 콜백함수의 recv_userheader 파라미터에 설정된다.

Sample

```
1 String sUserHeader = "USER_HEADER_DATA";  
2  
3 xDataSet.setUserHeader(xUserHeader);  
4 xDataSet.returnData();
```

See Also

returnData, getUserHeader

replaceHTMLTagFilter

Prototype

```
static String replaceHTMLTagFilter(String strData);
```

Parameter

Type	Name	설명
String	strData	<ul style="list-style-type: none"> 크로스 사이트 스트리핑 처리로 인해 치환된 XDataSet데이터

Return Value

Type	Name	설명
String	strData	<ul style="list-style-type: none"> 재치환된 XDataSet데이터

Description

Request 용 데이터 셋 문자열이 크로스 사이트 스트리핑 처리로 인해 치환되어 있는 경우, 재치환 하는 함수이다. 재치환 하는 문자는 다음의 5 가지이다.

- < : <
- > : >
- & : &
- ' : '
- " : "

Sample

```

1 String input = "&lt;script&gt;alert(&apos;XSS&apos;);&lt;/script&gt;";
2 String output = XDataSet5.replaceHTMLTagFilter(input);
3 System.out.println(output); // <script>alert('XSS');</script>
4
```

See Also

getData